The Performance Evaluation Research Center (PERC)

Participating Institutions:

Argonne Natl. Lab.	Univ. of California, San Diego
Lawrence Berkeley Natl. Lab.	Univ. of Illinois
Lawrence Livermore Natl. Lab.	Univ. of Maryland
Oak Ridge Natl. Lab.	Univ. of Tennessee, Knoxville

Website: http://perc.nersc.gov

PERC Overview

- An “Integrated Software Infrastructure Center” (ISIC) sponsored under DoE’s SciDAC program.
- Funding: approx. $2.4 million per year.
- Mission:
  - Develop a science of performance.
  - Engineer tools for performance analysis and optimization.
- Focus:
  - Large, grand-challenge calculations, especially SciDAC application projects.

Specific Objectives

- Understand key factors in applications that affect performance.
- Understand key factors in computer systems that affect performance.
- Develop models that accurately predict performance of applications on systems.
- Develop an enabling infrastructure of tools for performance monitoring, modeling and optimization.
- Validate these ideas and infrastructure via close collaboration with DOE Office of Science and others.
- Transfer the technology to end users.

Anticipated Benefits

Consider the economic value of improving the performance of a single high-end scientific application code by 20%.

Assume:

- $10 million computer system lease cost per year.
- $10 million per year in site costs, support staff, etc.
- 10-year lifetime of code.
- Code uses 5% of system cycles each year.

Savings: $2,000,000.

Scientific benefit (additional computer runs and research) is probably much higher.

Anticipated Benefits, cont.

- We rely heavily on commercial vendors for high-performance computer systems.
- We are invited by vendors to provide guidance on the design of current and future systems.

BUT

- At present we can provide only vague information – little if any quantitative data or rigorous analysis.

The performance monitoring and modeling capability to be developed in PERC will significantly improve our ability to influence future scientific computer systems.

Current State-of-the-Art

- Some tools collect performance data, but
  - They are not targeted to large parallel systems.
  - They are not able to collect performance data at individual levels of deep memory hierarchies.
- A few performance modeling techniques have been developed, but
  - They are time-consuming to generate, difficult to use, or have limited accuracy.
- Some automatic tuning techniques have been developed, but
  - They have been applied only in limited algorithm domains.
  - There is no hardened support for real-time optimization.

New Capabilities

- **Better Benchmarks:**
  - Polished, concise versions of real user codes, representing strategic application areas.
  - Kernel benchmarks extracted from real codes reduce complexity of analyzing full-size benchmarks.
  - Low-level benchmarks measure key rates of data access at various levels of memory hierarchy.
- **Modern performance monitoring tools:**
  - Flexible instrumentation systems capture hardware and software interactions, instruction execution frequencies, memory reference behavior, and execution overheads.
  - An advanced data management infrastructure tracks performance experiments and data across time and space.

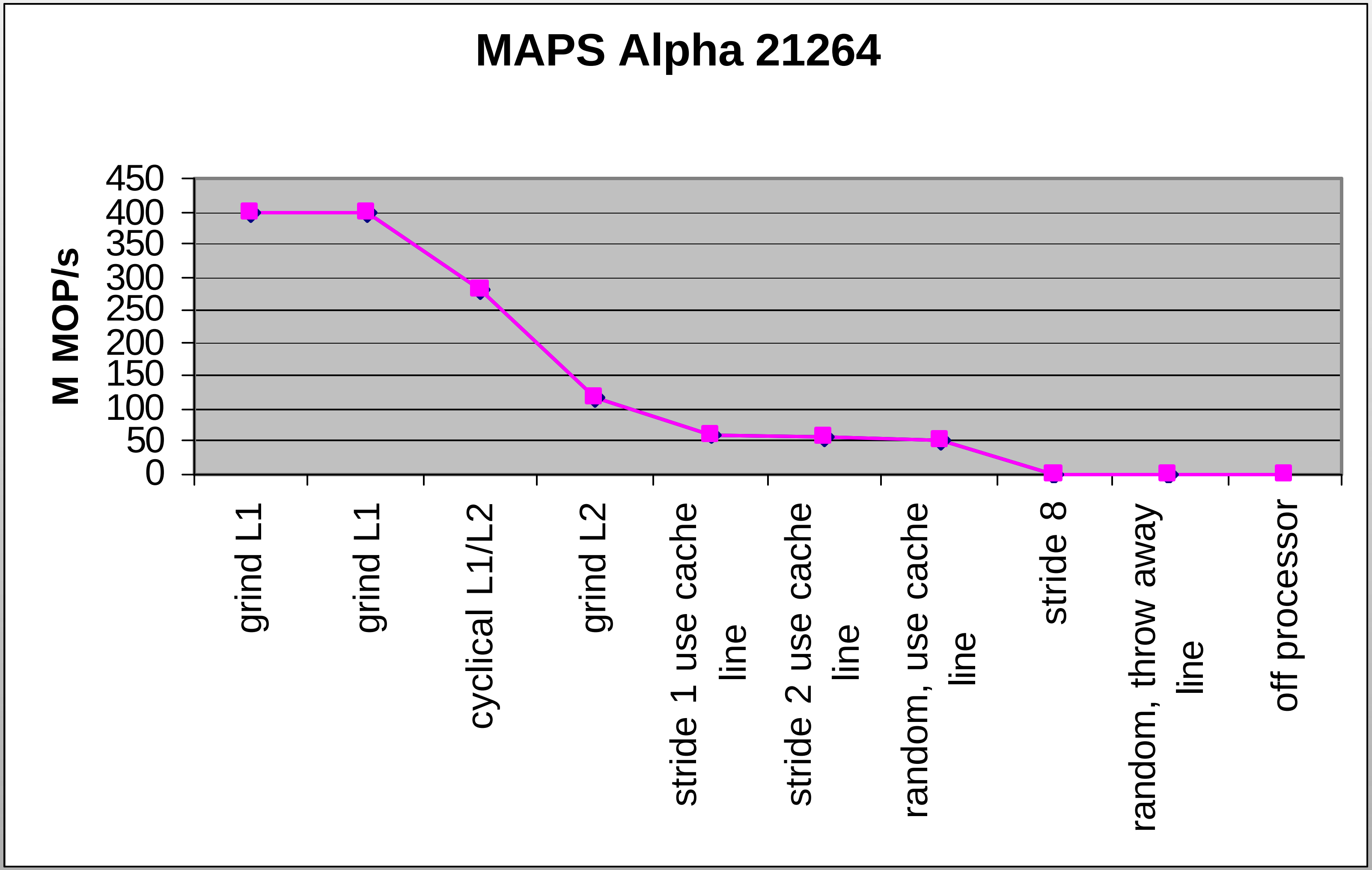
New Capabilities, cont.

- **Performance modeling:**
  - *Application signature* tools characterize applications independent of the machine where they execute.
  - *Machine signature* tools characterize computer systems, independent of the applications.
  - *Convolution* tools combine application and machine signatures to provide accurate performance models.
  - *Statistical models* find approximate performance models based on easily measured performance data.
  - *Phase models* analyze separate sections of an application, providing overall performance predictions.
  - *Performance bound* tools determine ultimate potential of an application on a given system.

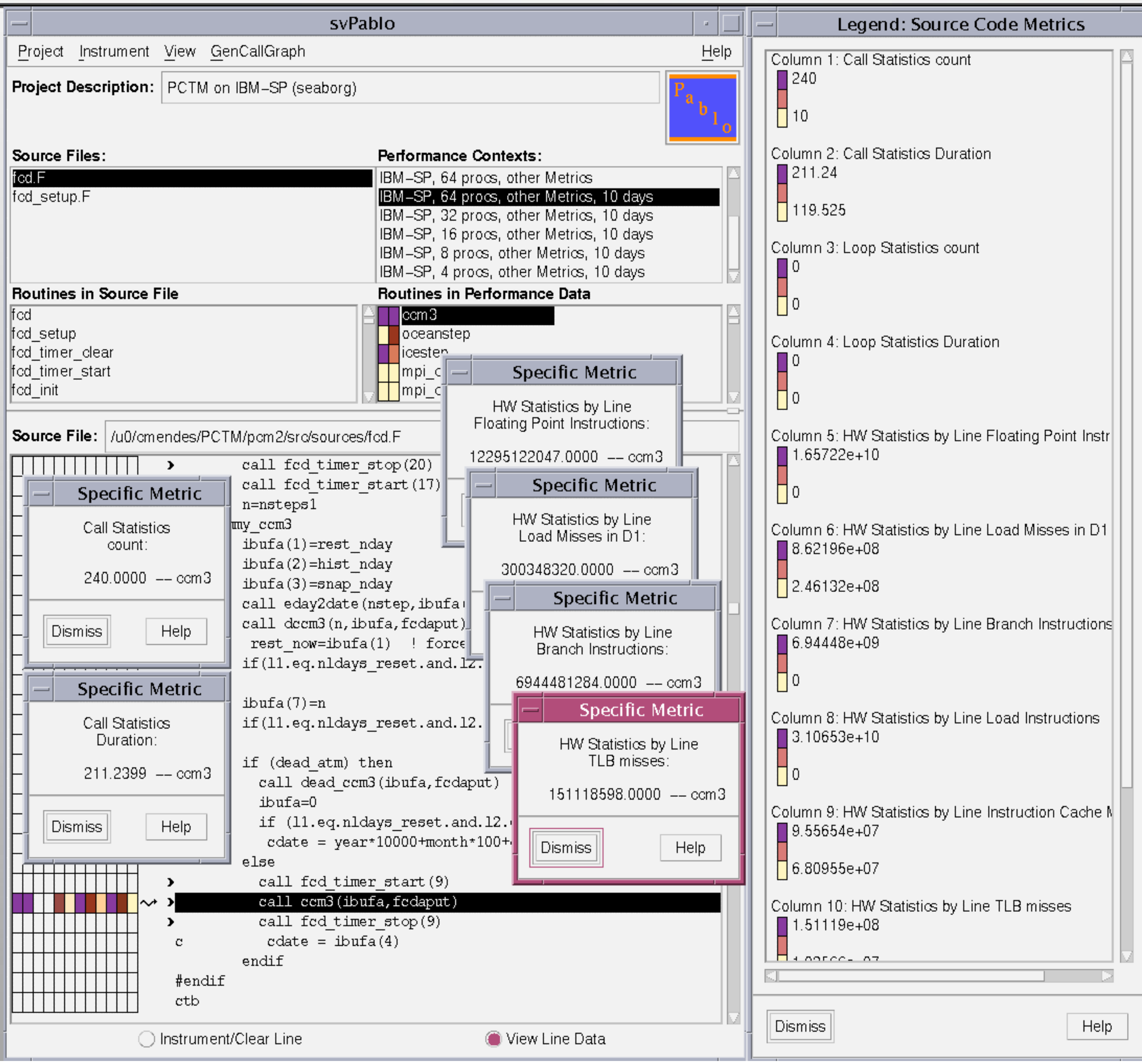
New Capabilities, cont.

- **Performance optimization:**
  - *Compile-time optimization* mechanisms analyze source code to improve performance.
  - *Self-tuning software* automatically tunes code based on real-time measurements of hardware environment.
  - *Performance assertions* permit user-specified run-time tests to possibly change the course of the computation depending on results.
  - *Performance portability* programming techniques to insure that code runs at near-optimal performance across a variety of modern systems.

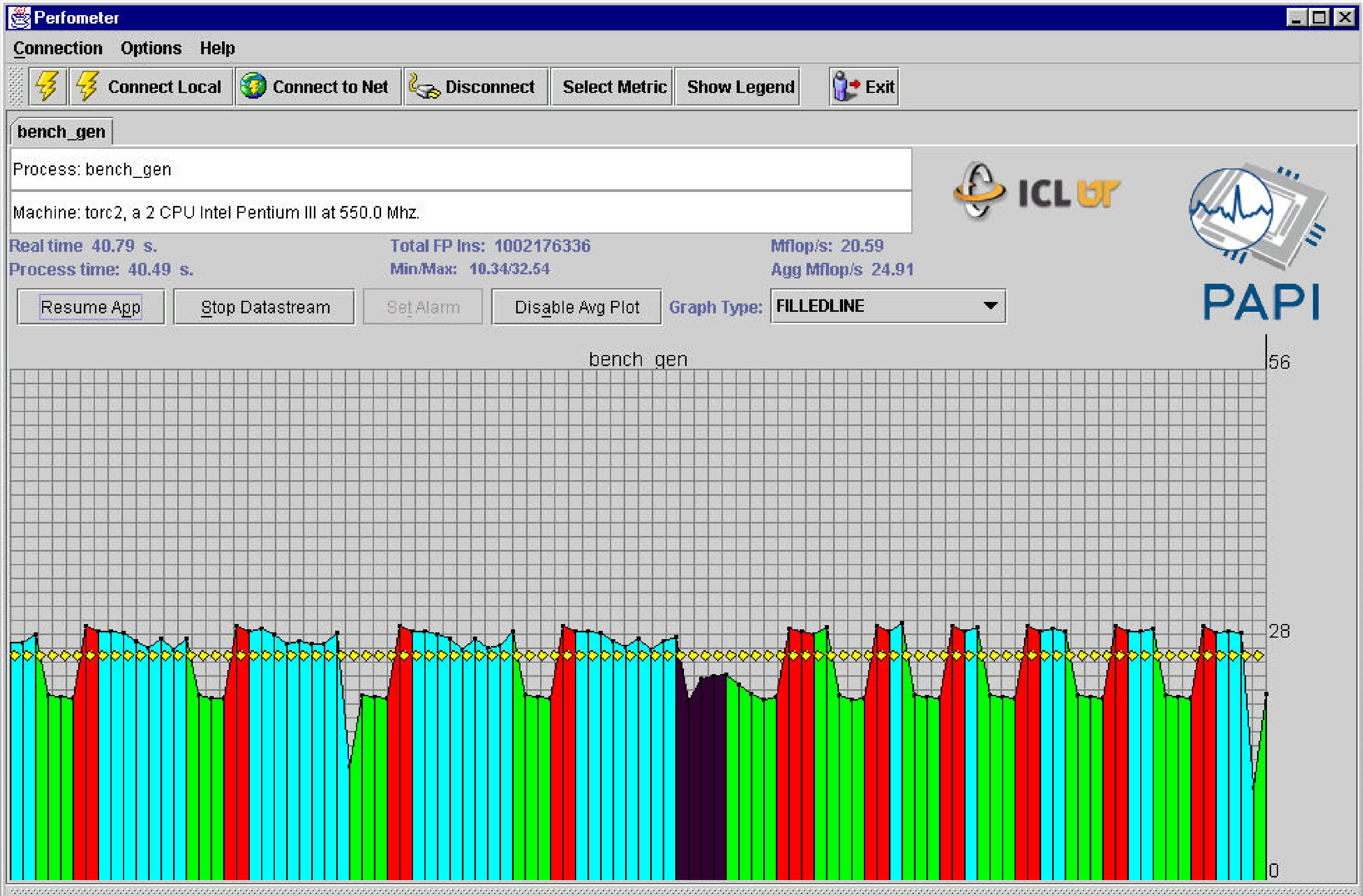
Measured Memory Access Patterns



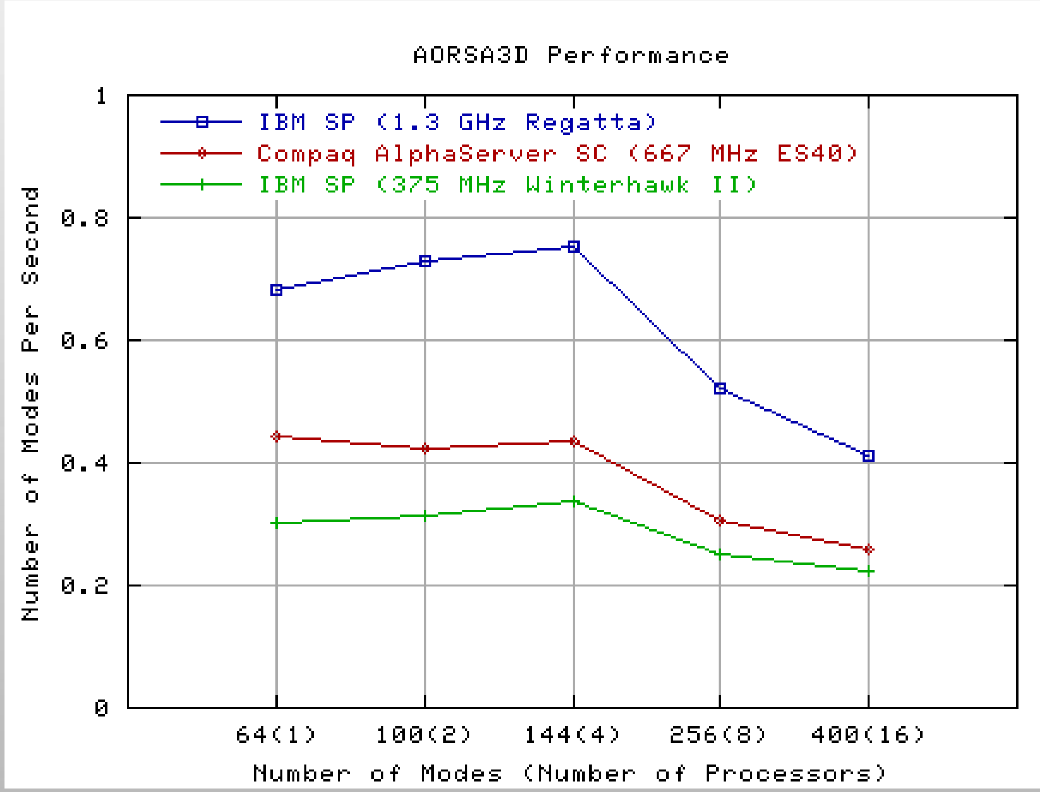
SvPablo Graphical Interface



PAPI Perfometer Interface



Performance Study of SciDAC Application



SciDAC Interactions

Codes have been acquired from these projects:

- Terascale Simulation of Neutrino-Driven Supernovae
- Advanced Computing for 21st Century Accelerators
- National Computational Infrastructure for Lattice Gauge Theory
- Collaborate Design and Development of Community Climate System Model for Terascale Computers
- Numerical Computation of Wave-Plasma Interactions
- Accurate Properties for Open-Shell States of Large Molecules
- Terascale Optimal PDE Solvers
- An Algorithmic and Software Framework for PDEs

Working with PERC

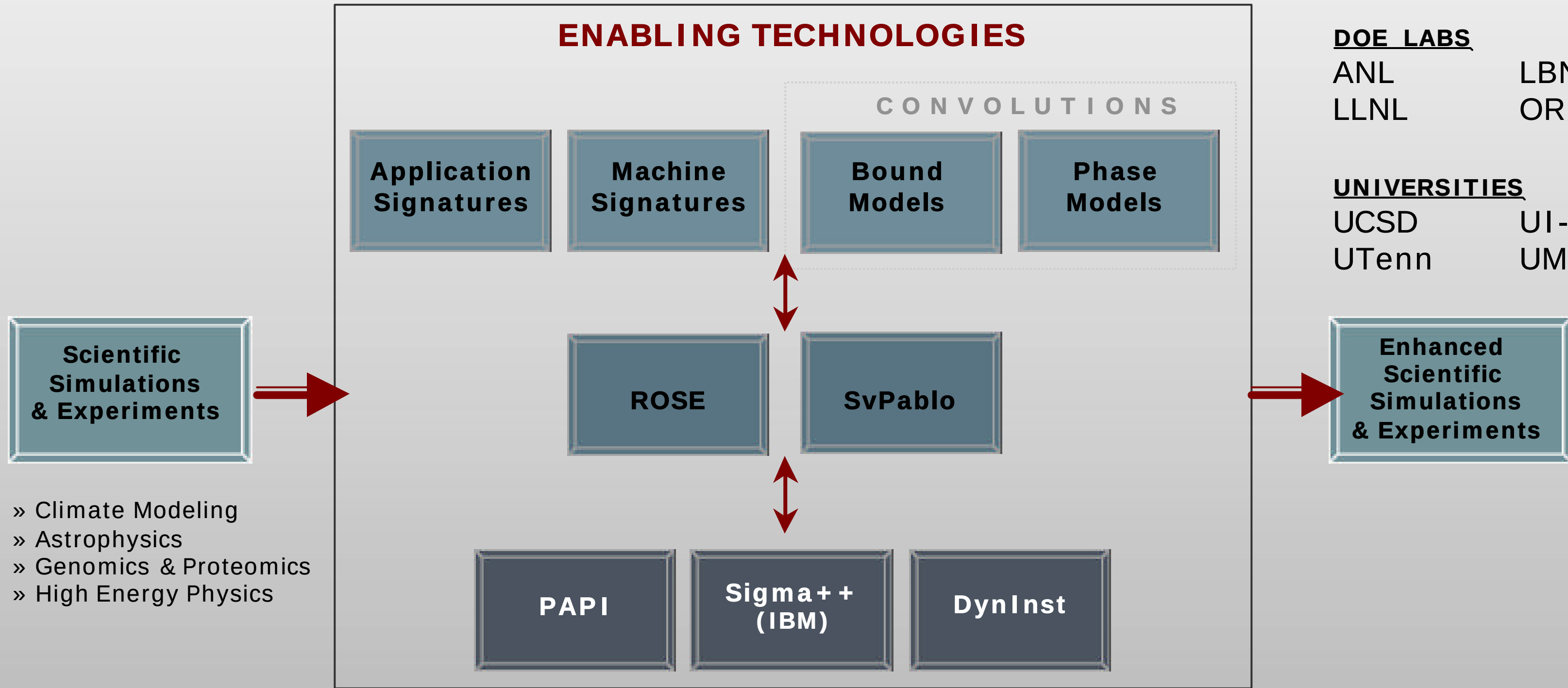
- Benchmarking
  - Application group works with PERC to specify relevant benchmark codes and problems.
  - PERC characterizes performance, generates performance models, and suggests optimizations.
- Performance Tools
  - PERC trains application developers to use tools.
  - Application group uses tools in development, providing feedback on functionality and future development

For further information: http://perc.nersc.gov

Summary

- Achieving optimal performance on HPC systems has compelling economic and scientific rationales.
- Performance is poorly understood – in depth-studies do not exist except in a handful of cases.
- PERC will pursue “performance science” and “performance engineering”, including improved benchmarks, monitoring tools, modeling techniques, and optimizers.

Developing a *science* for understanding performance of scientific applications on high-end computer systems, and *engineering* strategies for improving performance on these systems.



DOE LABS

ANL LBNL
LLNL ORNL

UNIVERSITIES

UCSD UI-UC
UTenn UMD

» Climate Modeling
» Astrophysics
» Genomics & Proteomics
» High Energy Physics

GOALS

Optimize and Simplify:

- Profiling of real applications
- Measurement of machine capabilities (emphasis on memory hierarchy)
- Performance prediction
- Performance monitoring
- Informed tuning

Understand the key factors in applications that affect performance.
Understand the key factors in computer systems that affect performance.
Develop models that accurately predict performance of applications on systems.
Develop an enabling infrastructure of tools for performance monitoring, modeling and optimization.
Validate these ideas and infrastructure via close collaboration with DOE SC and other application owners.
Transfer the technology to end-users.